

**Kimmo Hollanti**

**PELASTUSTOIMEN TEHTÄVÄNÄYTTÖSOVELLUKSEN  
KEHITTÄMINEN**

**Opinnäytetyö**

**KESKI-POHJANMAAN AMMATTIKORKEAKOULU**

**Tietotekniikan koulutusohjelma**

**Joulukuu 2010**

**TIIVISTELMÄ OPINNÄYTETYÖSTÄ**

<b>Yksikkö</b> Tekniikan ja liiketalouden yksikkö	<b>Aika</b> Joulukuu 2010	<b>Tekijä/tekijät</b> Kimmo Hollanti
<b>Koulutusohjelma</b> Tietotekniikan koulutusohjelma		
<b>Työn nimi</b> Pelastustoimen tehtävänäyttösovelluksen kehittäminen		
<b>Työn ohjaaja</b> DI Sakari Männistö		<b>Sivumäärä</b> 34 + 2
<b>Työelämäohjaaja</b> DI Jukka Kangasvieri		
<p>Opinnäytetyönä kehitettiin prototyyppi tehtävänäyttösovelluksesta pelastustoi- melle. Sovellusta käytetään pelastusalan henkilöstön informoimiseen hälytys- tehtävän sisällöstä sekä yksiköiden tilatiedoista. Sovellus kehitettiin automaatti- seksi ja helpoksi käyttää.</p> <p>Sovelluksen tarvitsemat tiedot tehtävistä ja muuttuneista tilatiedoista saadaan viestien välityksellä. Hätäkeskusten antamat tehtävät ja yksiköiden tilatiedot saapuvat TETRA-päätelaitteeseen tekstiviesteinä. Puhelimen ohjaamiseen ja tietojen siirtämiseen käytetään sarjaporttiyhteyttä.</p> <p>Sovellus ohjelmoitiin C++:lla käyttäen Qt:n kirjastoja ja kehitysympäristöä. Teh- tävänäyttösovellus todettiin käyttötarkoituksessaan toimivaksi ja se kehitettiin prototyyppiluonteesta huolimatta lähes valmiiksi tuotteeksi.</p>		
<b>Asiasanat</b> pelastustoimi, Qt, TETRA-verkko		

## ABSTRACT

<b>CENTRAL OSTROBOTHNIA UNIVERSITY OF APPLIED SCIENCES</b>	<b>Date</b> December 2010	<b>Author</b> Kimmo Hollanti
<b>Degree programme</b> Information Technology		
<b>Name of thesis</b> Developing a Task Monitor Software for Rescue Services		
<b>Instructor</b> Sakari Männistö		<b>Pages</b> 34 + 2 Appendices
<b>Supervisor</b> Jukka Kangasvieri		
<p>In this thesis a prototype of display application for fire and rescue services was developed. The application is used for rescue personnel to inform on the incoming fire or rescue tasks, and for the unit status information. The application was developed to be automatic and easy to use.</p> <p>The application gets required information on tasks and changed status information through messages. When ERG gives tasks, they send text messages to the TETRA-phone. The status information comes also to the phone. A serial port connection is used for controlling the phone and transferring data to the computer.</p> <p>The application was programmed in C++ and by using Qt libraries. The task monitor application was found to be useful, and despite it was a prototype it was developed to be like a finished product.</p>		
<b>Key words</b> Qt, rescue work, TETRA-network		

## **TIIVISTELMÄ**

## **ABSTRACT**

## **SISÄLLYS**

<b>1 JOHDANTO</b>	<b>1</b>
<b>2 LÄHTÖKOHDAT JA ONGELMAT</b>	<b>3</b>
<b>3 TYÖKALUT</b>	<b>6</b>
3.1 TETRA-verkko	6
3.2 PEI-rajapinta	7
3.3 Tehtäväviestit	8
3.4 Qt	9
<b>4 SUUNNITTELU</b>	<b>12</b>
4.1 Määrittely	12
4.1.1 Käyttötapaukset	13
4.1.2 Virheiden ja erikoistapausten sieto	15
4.2 Luokkakaavio	16
4.3 Tietokanta	17
4.4 Käyttöliittymä	18
<b>5 TOTEUTUS</b>	<b>21</b>
5.1 Käytetyt Qt-luokat	21
5.2 MySql-tietokanta	22
5.3 Sarjaporttiyhteys	22
5.4 Saapuvan datan käsittely	24
5.5 Kartta	24
5.5.1 Google Maps ja OpenStreetMap	25
5.5.2 Ongelmat	26
5.6 Testaus	27
5.7 Kuvia valmiista tehtävänäyttösovelluksesta	28
<b>6 POHDINTA</b>	<b>32</b>
<b>LÄHTEET</b>	<b>34</b>
<b>LIITTEET</b>	

## 1 JOHDANTO

Opinnäytetyössä kehitettiin tehtävänäyttösovellus, joka on osa pelastustoimen tietotekniikkaprojektia (PATI). Työssä pyrittiin kehittämään prototyyppisovellus, joka tarjoaa ratkaisun pelastuslaitoksen henkilöstön nopeaan ja helppoon informoimiseen tehtävistä ja yksiköiden tilatiedoista. Sovelluksen tarjoama informaatio esitetään visuaalisesti kosketusnäytöllä.

Pelastustoimen projektin aikana pyritään myös etsimään ratkaisuja muihin ongelmiin, kuten esimerkiksi palokaluston etävalvontaan, mutta opinnäytetyö rajoittuu vain tehtävänäyttöön ja tilatietoihin. Tehtävänäyttösovelluksen kehityksessä otettiin huomioon sovelluksen jatkokehitys ja mahdolliset lisäosat. Sovelluksen odotetaan edistävän paloturvallisuutta sekä kasvattavan pelastustoimen tehokkuutta ja toimintavarmuutta. Sovellus parantaa hälytystehtävään liittyvän informaation välittämistä tehtävään lähtevälle henkilöstölle. Sovelluksen selkeällä käyttöliittymällä ja automatisoidulla viestien käsittelyllä helpotetaan merkittävästi informaation käsittelyä.

Sovelluksen, yksiköiden ja hätäkeskuksen väliseen tiedonsiirtoon käytetään viranomaisten käyttämän Terrestrial trunked radio –verkon (TETRA) tarjoamia tekstiviestipalveluja. TETRA-päätelaitteena käytettiin Nokian valmistamaa laitetta. Suomessa viranomaisten käyttämä tietoliikenneverkko pohjautuu TETRA-standardiin. Viestien lukemiseen päätelaitteesta käytetään sarjaporttiyhteyttä ja AT-komentoja.

Tehtävänäyttö ohjelmoitiin Nokian Qt-kehitysympäristössä C++-ohjelmointikielellä. Qt:n valintaan vaikuttivat sen avoimuus, alustariippumattomuus ja asema uutena vartenotettavana ohjelmointiympäristönä. Qt antoi myös mahdollisuuden opiskella ja tutustua syvemmin sen tarjoamiin mahdollisuuksiin. Ohjelmassa käytettävä tietokanta toteutettiin MySQL:llä.

Sovellus suunniteltiin ja räätälöitiin kokonaisuudessaan Keski-Pohjanmaan ja Pietarsaaren alueen pelastuslaitoksen Kokkolan paloaseman tarpeisiin, mutta se voi-

daan ottaa käyttöön muillakin asemilla. Opinnäytetyön tuotteena syntynyt ohjelma tulee olemaan päivittäisessä käytössä pelastuslaitoksella. Sen mahdollisesta jatkokehityksestä vastaavat Keski-Pohjanmaan ammattikorkeakoulu ja muut PATI-projektin yhteistyökumppanit. Tietoturvasyistä liitteet ovat salaisia.

## 2 LÄHTÖKOHDAT JA ONGELMAT

Pelastustoimessa on velvoitteena tuoda apua mahdollisimman nopeasti sitä tarvitseville. Ohjeena on, että tehtävän saatuaan tehtävään kiinnitetyt yksiköt, esimerkiksi paloautot, ovat yhden minuutin sisällä jo matkalla kohteeseen. Aikarajan sisällä henkilöstö valmistautuu tehtävään. Olennaisena osana valmistautumista on selvittää tulevan tehtävän luonne ja kohteen osoite.

Yksiköiden hälyttäminen tehtävään tapahtuu hätäkeskuksesta käsin. Hälytys annetaan kuulutuksena, faksina, viesteinä TETRA-päätelaitteisiin ja johtamisjärjestelmän kautta. Ongelmiksi ovat muodostuneet hätäkeskusten kuulutusten heikko äänenlaatu, faksin hitaus ja TETRAn viestipalveluiden rajoitteet. Viestien koko on rajoitettu, ja päätelaitteet eivät pääsääntöisesti osaa yhdistellä useita viestejä kokonaisuuksiksi. Tehtävistä saatavan informaation käytettävyys on huono, mikä vaikeuttaa toimintaa ja siihen valmistautumista. Päätelaitteen naputtelu pukeutumisen aikana on vaikeaa. Ongelmien vuoksi on muodostunut tarve tehtävänäyttösovellukselle, jossa tehtävän ja kohteen tiedot on esitetty selkeästi koostettuna.

Pelastustoimella on käytössä pelastusalan kenttäjohtamisjärjestelmä (PEKE). Tämä järjestelmä on kehitetty poliisien kenttäjohtamisjärjestelmän (POKE) kanssa rinnakkain. PEKE-järjestelmä ei sovellu kuitenkaan sellaisenaan yllä mainittuun tarkoitukseen ilman suuria muutoksia. Täten tehtävänäyttösovelluksen kehittäminen on perusteltua ja tarpeellista.

Tehtävänäyttösovelluksen ei kuitenkaan ole tarkoitus toimia pääasiallisena tai ainoana tiedon lähteenä. Sovelluksen monitorit sijoitetaan paloaseman kalustohalliin, henkilöstön pukeutumistilan seinälle. Näistä monitoreista henkilöstö voi tarkastella kohteesta annettuja tietoja varustautuessaan tehtävään. Varustautuminen eli pukeutuminen ja ajoneuvoon nousu edellyttää molempien käsien käyttöä, jolloin käsikäyttöiset laitteet eivät ole käytännöllisiä. Tarvitaan siis vaihtoehtoinen ja itsenäisesti toimiva informaatiojärjestelmä. Tehtävänäytöstä ilmenevät tehtävän suorittamisen kannalta kaikki tarpeelliset tiedot helposti ja visuaalisesti esitettynä.

Tehtävänäyttösovellukseen on yhdistetty myös tilatiedot yksiköistä eli kulkuneuvoista ja niiden miehistöistä. Tilatiedot kertovat jokaisen yksikön reaaliaikaisen tilan, esimerkiksi sen, onko yksikkö asemalla, matkalla vai kohteessa. Niiden avulla lähetetään myös yksikön vahvuustietoja. Käyttäjä lähettää tilatiedot manuaalisesti TETRA-päätelaitteella. Tiedot auttavat yksiköiden paikantamisessa ja johtamisessa.

Tilatietoja voi seurata reaaliaikaisesti tehtävänäytön avulla. Tehtävän sattuessa ovat nähtävillä tehtävään kiinnitetyt yksiköt ja niiden tilat. Tehtävänäytöltä voi seurata myös sairaankuljetusajoneuvojen tiloja. Tämä on hyödyllistä etenkin silloin, kun tehtävään on hälytettynä myös sairaankuljetusyksiköitä.

Tehtävänäyttösovelluksesta kehitetään mahdollisimman helppokäyttöinen ja vähän käyttäjältä toimenpiteitä vaativa. Tehtävän tullessa tietojen on ilmaannuttava automaattisesti näytölle ja tehtävään liittyvien yksiköiden tilatiedot on voitava nähdä yhdellä vilkaisulla. Tehtävätietojen on poistuttava näytöltä tietoturvasyistä automaattisesti tietyn ajan kuluttua. Ohjelma ei siis välttämättä vaadi käyttäjältä mitään toimia. Käyttäjä voi kuitenkin tarvittaessa olla vuorovaikutuksessa ohjelmaan kosketusnäytön avulla. Kuviossa 1 on kuvattuna tehtävänäyttösovelluksen toimintaympäristö. Kuviosta nähdään että kosketusnäyttö sijaitsee naulakoiden välittömässä läheisyydessä





KUVIO 1. Tehtävänäyttösovelluksen sijainti kalustohallissa

### 3 TYÖKALUT

Tehtävänäyttösovellusta kehitettäessä on hyödynnetty tekniikkaa monelta eri osalta. Suomessa on varsin kattava TETRA-standardiin perustuva viranomaisverkko, jonka mahdollistamat viestipalvelut ovat tärkeä osa sovellusta. TETRA-päätelaitteen ja tietokoneen sarjaporttiliitännät mahdollistavat kommunikoinnin laitteiden välillä. Varsinainen sovelluksen ohjelmointiin tietokoneelle käytetään C++:aa ja Qt-kirjastoja.

#### 3.1 TETRA-verkko

Terrestrial trunked radio eli TETRA-standardiin perustuvaa viranomaisverkkoa käytetään Suomessa viranomasten tietoliikenteeseen. Se on European Telecommunications Standardisation Institute:n (ETSI) määrittelemä standardi. TETRA on pääpiirteissään hyvin samankaltainen kuin tunnetumpi global system for mobile communications -järjestelmä (GSM). TETRA perustuu suurimmaksi osaksi samaan tekniikkaan kuin GSM, mutta joiltakin osin ne eroavat toisistaan merkittävästi. TETRAn taajuusjako on 25 kHz. Se on myös yleisöltä suljettu ja salattu verkko. TETRA on niin sanottu trunking-verkko, eli sen resurssit ovat kaikkien verkkoon kytkettyjen käyttäjien käytettävissä. (Penttinen 2006, 39–41.)

Suomessa on käytössä TETRAn perustuva viranomaisverkko (VIRVE), jota käyttävät muun muassa poliisi-, pelastustoimi. Verkon palveluja käytetään TETRA-radiolla, joka muistuttaa paljon ulkonäöltään ja toiminnallisuudeltaan tavallisia matkapuhelimia. Suomessa käytetään suurimmaksi osaksi EADSin ja Sepuran valmistamia radioita tai päätelaitteita. (Heikkonen, Pesonen & Saaristo 2005, 34.)

TETRAssa on puhepalvelujen lisäksi GSM:stä tuttuja tekstiviestipalveluja. Short Data Service (SDS) on TETRA-verkon lyhytsanomapalvelu, joka vastaa yleisesti tunnetumpaa GSM:n Short Message Serviceä (SMS). SMS-viesteihin verrattuna viestit kulkevat TETRA-verkossa lähes viiveettä, joten verkossa ei kulje vanhentu-

nutta tietoa. Viestejä on kaksi eri tyyppiä: statusviestejä ja tekstiviestejä. (Heikkonen ym. 2005, 47–49.)

TETRA:n statusviestit ovat lyhyitä viestejä, jotka koostuvat vain pelkästä numero-koodista. Koodien merkitys on ennalta sovittu. Statusviestejä voidaan käyttää esimerkiksi tilatietiedon välittämiseen päivystäjälle tai muille verkon käyttäjille. Statusviestit kuormittavat verkkoa erittäin vähän, ja ne ovat tärkeitä yksiköiden johtamisen kannalta. (Heikkonen ym. 2005, 47–49.)

Tekstiviestejä voi lähettää samaan tapaan kuin statusviestejä. Viesteillä annetaan yleensä yksityiskohtaisia tietoja, kuten osoitteita, nimiä tai potilastietoja. Tekstiviestit ovat tehokas ja luotettava tapa lähettää tietoa verkon käyttäjille. Viestit tallentuvat päätelaitteeseen, ja niitä voidaan tarkastella useampaankin kertaan. (Heikkonen ym. 2005, 50.)

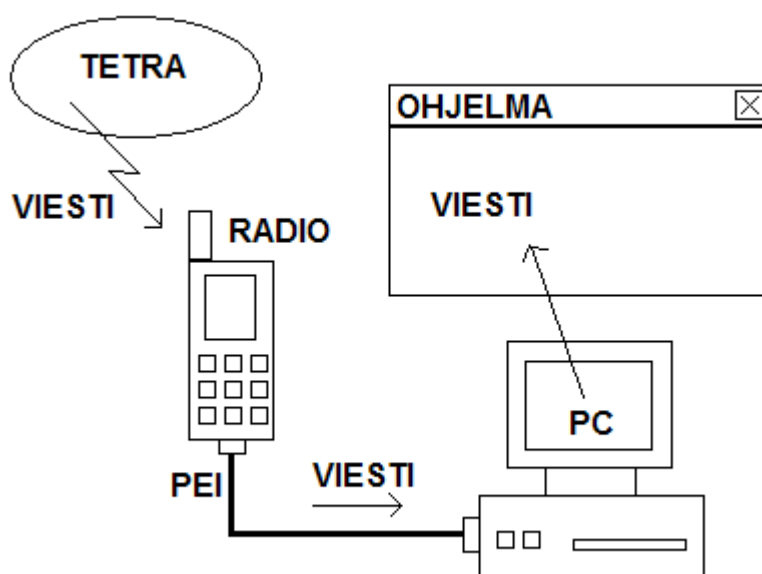
GSM:stä poiketen viestejä ei tallenneta myöhempää välitystä varten. Tämä tarkoittaa, että viestejä voi jossain tapauksissa kadota lähetysvaiheessa. Viestejä voi lähettää myös monille eri vastaanottajille samaan aikaan. Tällöin käytetään puheryhmiä. TETRA-päätelaitteen voi asettaa kuuntelemaan tiettyjä puheryhmiä. TETRA-viestien tai sanomien lähettäjät voidaan yksilöidä laitteisiin ohjelmoitujen yksilötunnusten eli Individual TETRA Subscriber Identity -tunnuksen (ITSI) avulla. ITSI koostuu numeroista. (Heikkonen ym. 2005, 44–50.)

### 3.2 PEI-rajapinta

TETRA-radioihin voidaan liittää oheislaitteita, kuten tietokoneita, peripheral equipment interface -rajapinnan (PEI) avulla. Fyysisesti rajapinta voidaan toteuttaa esimerkiksi sarjaporttiliitännän (RS-232) avulla. PEI-rajapintaa käyttäen puhelinta voi ohjata tietokoneella. Ohjaukseen käytetään modeemeista tuttuja AT-komentoja. Niiden avulla voidaan esimerkiksi lähettää ja lukea viestejä. AT-komennot ovat yksinkertaistettuna muotoa: *at+komento*. Kun komento lähetetään sarjaportin kautta TETRA-laitteelle, se reagoi komennon mukaisesti ja vastaa lähettämällä kuittauk-

sen tai halutut tiedot tietokoneelle. (Penttinen 2006, 43–48; Heikkonen ym. 2005, 59.)

PEI-rajapintaa käytetään tehtävänäyttösovelluksessa välittämään tietoa sovelluksen ja TETRA-päätelaitteen välillä. Päätelaite mahdollistaa vuorostaan tiedonsiirron hätäkeskuksesta päätelaitteeseen ja toisin päin puhelimesta TETRA-verkkoon. Kuviossa 2 on kuvattu PEI-rajapinnan keskeisimmät laitteet ja käsitteet. Kuviosta ilmenee myös, kuinka tietoa voidaan siirtää TETRA-verkosta tietokoneeseen.



KUVIO 2. PEI-rajapinnan käyttö tiedonsiirrossa

### 3.3 Tehtäväviestit

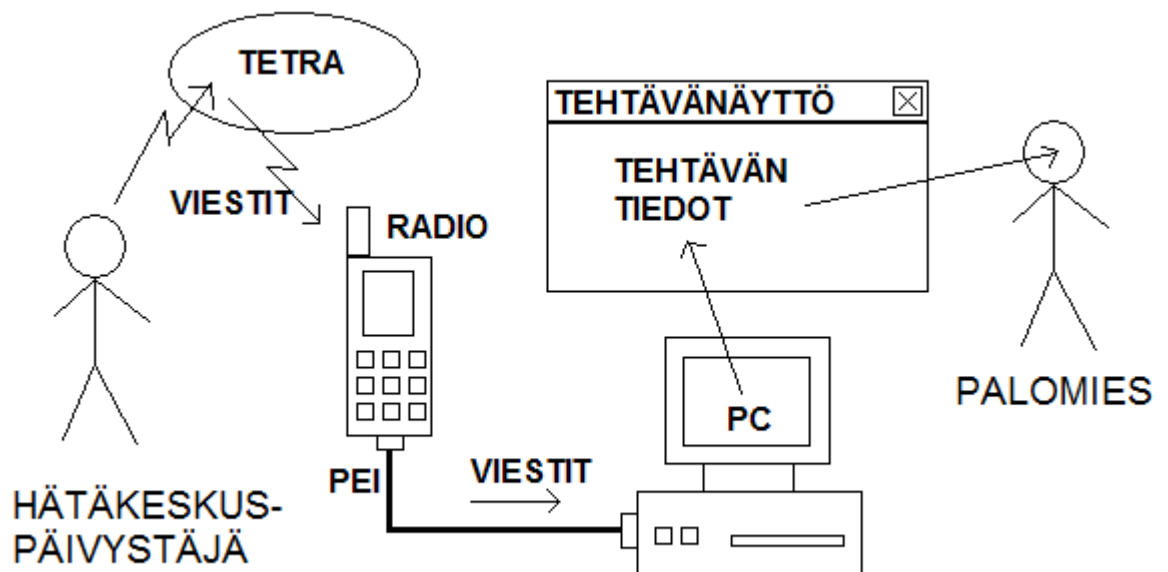
Yksiköiden tilatietoja voi vastaanottaa liittymällä tiettyyn ryhmään. Hätäkeskus lähettää tehtäväviestejä tiettyjen TETRA-päätelaitteiden lisäksi myös PEKE-palvelimille, josta niitä voi tarkastella PEKE-sovelluksella. TETRA-päätelaitteisiin tulevat viestit eroavat PEKE-viesteistä jonkin verran. Tehtävänäyttösovelluksessa käytetään hyväksi näitä samoja viestejä.

Tehtäväviestit sisältävät tietoja kohteesta, ilmoittajasta ja tehtävän luonteesta. Viestejä lähetetään hätäkeskuksesta kolme kutakin tehtävää kohden. Viesteistä

ilmenee tehtävännumero, osoitetarkkuus, tehtävälaji, kiireellisyys, kunta, kadun nimi, talon numero, mahdollinen toinen kadun nimi, resurssit, koordinaatit, ilmoitus-aika ja ilmoittajan tiedot puhelinnumeroineen ja nimineen.

Liitteessä 1 on tarkemmin selvitetty viestien sisältö. Viestit koostuvat kentistä, jotka on eroteltu viesteissä toisistaan puolipisteillä. Kenttien maksimikoot on esitetty koko-sarakkeessa merkkeinä. TETRAssa viestien koko on rajoitettu, joten kenttienkin kokoa on täytynyt rajoittaa. Rakenteesta tai tietotyypeistä ei ole saatavilla tarkempaa määrittelyä, joten testausvaiheessa on tutkittava ja todettava itse kenttien rakenne.

Jos kuvioon 2 lisätään hätäkeskuksen päivystäjä ja palomies, voidaan kuvata koko viestintäketju päivystäjältä palomiehelle (KUVIO 3). Kuvio kertoo käytännössä sen, kuinka informaatio kulkee eri muodoissa tehtävän antajalta toteuttajalle.



KUVIO 3. Viestintäketju päivystäjältä palomiehelle

### 3.4 Qt

Qt on alun perin Trolltech Oy:n kehittämä alustariippumattomien ohjelmistojen kehitysympäristö. Nokia osti Trolltechin kesäkuussa 2008 ja yhdisti Trolltechin lippu-

laivan Qt:n omiin verkostoihinsa. Uuden yksikön nimeksi tuli Qt Development Frameworks. (Nokia 2009; Nokia 2010; Blanchette & Summerfield 2009, xix–xxi.)

Qt:lla voi luoda alustariippumattomia graafisia ja tekstipohjaisia käyttöliittymiä. Tuotettua lähdekoodia voidaan käyttää monissa eri ympäristöissä ilman merkittävää uudelleen kirjoittamista. Qt:lla voi tehdä ohjelmia Windowsille, Linuxille, Mac OS X:lle, Symbianille, Maemolle, Windows CE:lle ja Windows Mobile:lle. Qt sisältää kattavan luokkakirjaston, kehittyneen graafisen kehitysympäristön ja kääntäjän. Kehitysympäristö on nimeltään Qt Creator. (Nokia 2009; Nokia 2010.)

Qt:n tärkein ominaisuus on suhteellisen kattava luokkakirjasto, joka koostuu valmiista korkean tason luokista. Kirjasto tarjoaa valmiit luokat Internet-liikenteestä aina multimedian esittämiseen asti. Tämä vähentää ohjelmoijan työtä sekä lähdekoodin pituutta merkittävästi. Qt:n kattavan luokkakirjaston lisäksi käytettävissä on runsaasti vapaata lähdekoodia ja ilmaisia laajennuksia. Vapaa lähdekoodi on General Public License -lisenssin (GPL) alaista. Tämä tarkoittaa, että GPL:n alaista koodia voi hyödyntää omissa sovelluksissaan, jos sekin on saman lisenssin alainen.

Qt:n kehitysympäristö sisältää tekstieditorin lisäksi kääntäjän, Qt Designerin ja laajan apukirjaston ja dokumentaation. Ohjelmien graafiseen ulkoasuun ja käyttöliittymän suunnitteluun Designer antaa paljon hyödyllisiä ominaisuuksia. Dialogien tai muiden ikkunoiden suunnittelu on helppoa ja yksinkertaista. Komponentteja on valmiiksi kattava valikoima painonapeista fonttipudotusvalikoihin (QFontComboBox) asti. Komponenttien sijoittelu on helppoa ja suoraviivaista. Lisäksi on mahdollista laittaa automaattisia asetteluja, joiden avulla voi nopeasti asettaa komponentit paikoilleen. (Nokia 2009.)

Qt Creatoriin integroitu ohje on erittäin käytännöllinen ja tarpeellinen apu ohjelmoijalle. Ohjeessa on kaikista luokista ja niiden metodeista, konstruktoreista ja signaaleista kattava dokumentaatio. Ohjeelle on käyttöä, koska muun muassa nimeämiskäytännöt poikkeavat muista kehitysympäristöistä.

Qt tarjoaa mielenkiintoisen lähestymistavan olioiden vuorovaikutukseen. Luokat tarjoavat signaaleja, jotka emittoituvat, kun jotain tapahtuu. Esimerkiksi näppäinolio emittoi clicked-signaalin, kun näppäintä on painettu. Signaali voidaan yhdistää metodiin (slot), joka tällöin ajetaan. Yhdistämiseen käytetään connect-komentoa: `connect(pushButton, SIGNAL(clicked()), this, SLOT(close()))`; Esimerkkikoodi edellä sulkee kyseisen olion, johon this viittaa, kun nappia on painettu. (Blanchette & Summerfield 2009, 5–21.)

Lomakkeiden ja ikkunoiden komponenttien asettelun suunnittelu ja toteutus on Qt:lla helppoa. Ohjelmoijan avuksi on runsaasti tarjolla asetelmia eli layout:eja. Asetelmia käytettäessä komponentit asettuvat asetelman määrittelemiin paikkoihin ja niiden koot voivat muuttua. Asetelmien avulla komponentit asettuvat nopeasti silmää miellyttävään järjestykseen. Qt:n erikoisuutena voidaan pitää näkymättömiä jousia, spacereita, joiden avulla voi asetelmaan muodostaa suurempia välejä komponenttien välille. Toinen erikoisuus liittyy ikkunoiden koon muuttamiseen. Komponentit voidaan asettaa venyviksi, jolloin ne venyvät ikkunan kokoa muutettaessa. Tämä on mahdollista ilman käsin koodaamista, joten laadukkaita ohjelmia on helppo tehdä vähäisellä koodilla. (Blanchette & Summerfield 2009, 141–147.)

## 4 SUUNNITTELU

Suunnitteluvaiheessa määritellään ohjelman käyttötapaukset, toiminnot, vaatimukset, käyttöliittymä sekä tietokannan ja ohjelman rakenne pääpiirteissään. Ohjelmistokehityksen riskejä pystytään eliminoimaan, jos tärkeimmät ja kriittiset osat ohjelmasta tehdään ensin. Perinteisemmässä lähestymistavassa aluksi määritellään koko projekti tarkasti ja myöhemmin toteutetaan toiminnot.

Tärkeimpänä osana ohjelmaa on PEI-rajapinnan kehittäminen. Käytännössä tämä tarkoittaa, että sarjaporttia on kyettävä lukemaan ja siihen on pystyttävä myös kirjoittamaan. Seuraavaksi tärkeimpänä, ja myös vaikeimpana osana, on kartan soveltaminen käyttöliittymään. Kolmas vaihe on TETRA-päätelaitteen asetusten muokkaaminen tarkoituksenmukaisiksi ja laitteen ominaisuuksiin tutustuminen. Ohjelman pienempinä osina voidaan pitää näyttökomponenttien tutkimista ja soveltamista käyttöliittymään. Neljäntenä vaiheena on viestien koostaminen ja poimiminen tietovirrasta.

### 4.1 Määrittely

Tehtävänäyttösovelluksen tulisi olla yksinkertainen, luotettava, informaatiota lisäävä ja havainnollistava. Sovelluksen ei ole tarkoitus toimia tärkeimpänä tai päällimmäisenä tehtävistä ilmoittavana järjestelmänä. Tarkoitus on pikemminkin tuottaa lisäinformaatiota ja apuvälineistöä. Ohjelma ei siis ole pelastustoimen tai hälytystehtävien kannalta kriittisessä asemassa, mutta se nopeuttaa ja helpottaa pelastushenkilöstön valmistautumista tehtävään.

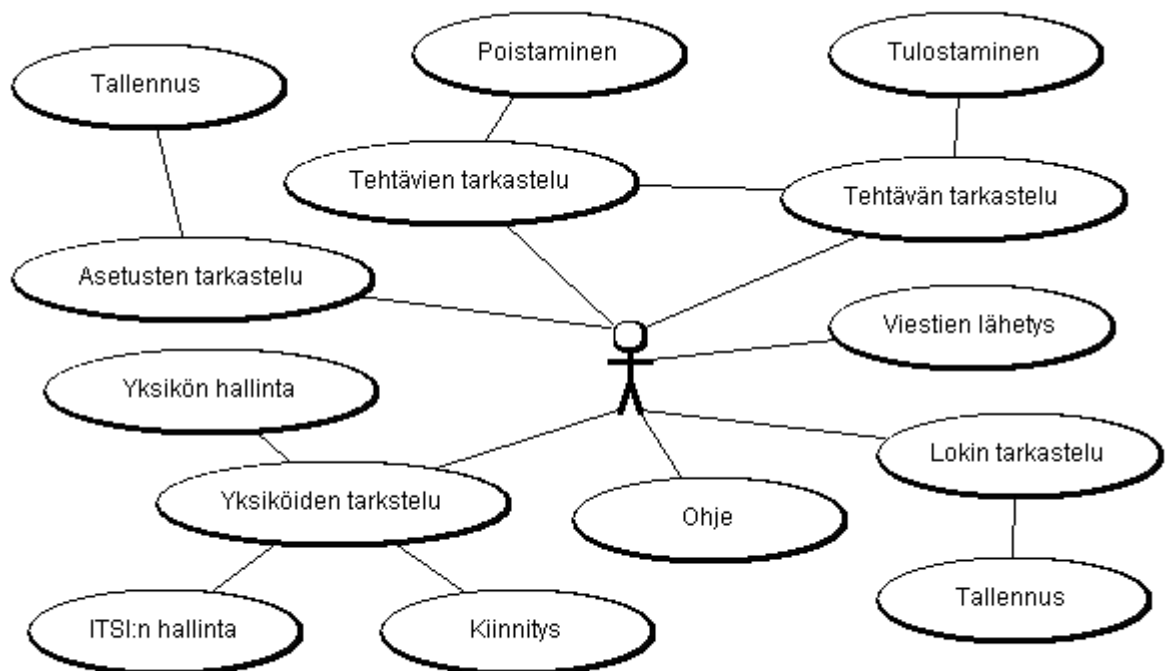
Ohjelman tulisi osata lähettää ja vastaanottaa viestejä sekä analysoida niitä. Jos ne vastaavat koostumukseltaan PEKE-viestejä, niistä poimitaan tiedot näytölle ja tietokantaan. Myös yksiköiltä saatavat tilatiedot tallennetaan tietokantaan. Ohjelma tallentaa kaiken saapuvan ja lähetetyn tiedon lokiin, josta tietoliikennettä voi tarkastella.



Ohjelman on oltava mahdollisimman käyttäjäystävällinen, ja sen on toimittava ilman käyttäjän aktiivista vuorovaikutusta. Tietosuojan vuoksi näytössä on näytönsäästäjä, joka peittää alleen tehtävänäyttösovelluksen. Tehtävän tullessa näytönsäästäjä poistuu ja tehtävänäyttöikkuna ilmestyy päällimmäiseksi peittäen pääikkunan ja kaikki muut mahdolliset ohjelmat tai ikkunat.

#### 4.1.1 Käyttötapaukset

Ohjelmaan tulee vain yksi käyttäjätaso, jolla on kaikki oikeudet käyttää toimintoja. Ohjelmaa kehitettäessä voidaan lisätä käyttäjiä, joilla on rajoitettuja oikeuksia käyttää ohjelmaa. Esimerkiksi vuoromestarilla, palomiehillä ja päällystöllä olisi tällöin omat profiilit ja käyttöoikeudet. Pääkäyttäjällä tulee olemaan seuraavia päätoimintoja käytettävissä: asetusten, lokin, tehtävien ja yksiköiden tarkastelu ja viestien lähettäminen. Kuviossa 4 on esitelty, miten eri käyttötapaukset nivoutuvat toisiinsa.



KUVIO 4. Käyttötapauskaavio

Jokaiselle pääkäyttötapaukselle tulee omat välilehtensä pääikkunaan. Tehtävät-välilehdellä voi tarkastella tehtäviä listalta, jossa tehtävät ovat aikajärjestyksessä. Jokaisen tehtävän kohdalla näkyvät tärkeimmät tiedot. Tehtävää klikattaessa saadaan esille ikkuna, jossa kaikki tehtävään liittyvät tiedot ovat näkyvillä. Ikkuna tulee olemaan täsmälleen sama kuin hälytystehtävän sattuessa automaattisesti ilmestyvä tehtävänäyttöikkuna. Asetuksissa annetun ajan kuluttua ikkuna sulkeutuu. Tehtäviä voi tarkastella tai poistaa myöhemminkin tehtävälialta.

Asetukset-välilehteen suunnitellaan mahdollisuus tarkastella ja muuttaa asetuksia. Toisin sanoen siinä on mahdollista muuttaa joitakin ohjelman toimintaan liittyviä asioita, kuten ikkunan sulkeutumisaikaa ja automaattista tulostamista. PEI-rajapinnan eli sarjaporttiliitännän sulkeminen ja aukaiseminen on myös mahdollista. Tietokannan valinta ja sen parametrit asetetaan myös tässä välilehdessä. Parametreja ovat tietokannan nimi, palvelin, käyttäjätunnus ja salasana.

Loki-välilehteen tallentuu kaikki PEI-rajapinnassa molempiin suuntiin siirtyvä dataliikenne. Tieto on tarkasteltavissa ohjelmasta suoraan tai tiedostosta käsin. Kaikki lokiin tallentunut tieto siirretään automaattisesti tiedostoon kovalevylle, kun vuorokausi vaihtuu. Lokia voidaan hyödyntää ongelma- ja virhetilanteiden sattuessa. Sinne tulevat tallentumaan myös ohjelman antamat virheilmoitukset. Loki-välilehden kautta käyttäjä voi tyhjentää lokin tai tallentaa sen tiedostoon. Sarjaporttiliikenteen tallentumista lokiin voi suodattaa jättämällä ylimääräinen liikenne kirjaamatta. Ylimääräistä liikennettä ovat muun muassa paikannustiedot ja toimialueeseen kuulumattomien yksiköiden tilatiedot.

Viestien lähetys ohjelman kautta toteutetaan erillisellä lomakkeella, joka sijoitetaan omaan välilehteensä. Lomakkeelle asetetaan vain vastaanottajan ITSI ja haluttu viesti. Viestien lähetys ei ole kovin tärkeä osa tehtävänäyttösovellusta, mutta se tarjoaa lisäarvoa ohjelman käytettävyydelle ja mahdollistaa lisäsovellukset.

Yksiköitä voi tarkastella samaan tapaan kuin tehtäviä. Yksiköt ovat listattuna taulukkoon, jossa on myös tärkeimmät yksikköä koskevat tiedot. Ne koostuvat nimestä, ITS:stä ja tilatiedosta. Välilehteen sijoitetaan vasemmalle yksikkötaulukko ja oikealle painonappeja. Painonappien avulla käyttäjä voi lisätä ja poistaa yksiköitä tai

ITSIä. Yksikön tietojen muuttaminen tapahtuu valitsemalla haluttu yksikkö ja painamalla muokkaa-nappia.

Yksiköiden kiinnittäminen tarkoittaa tiettyjen yksiköiden lisäämistä painonapeiksi pääikkunassa oikealla oleviin laatikoihin. Kun yksikkö on kiinnitetty laatikkoon, siitä muodostuu painonappi. Sen nimeksi tulee yksikön nimi ja taustaväriksi sen tilatiedon väri. Samalla tavalla tapahtuu myös sairaankuljetusyksiköiden kiinnittäminen. Ne näkyvät myös tehtävänäyttöikkunassa.

Jokaisella yksiköllä on yleensä yksi tai useampi TETRA-puhelin. Yksikön puhelinten ITSI:t asetetaan yksiköt-välilehdessä. Jos tietty ITSI on yhdistetty yksikköön, sovellus tulee osata yhdistää tulevan statusviestin kyseiseen yksikköön. Statusviestin saavuttua sovellus päivittää yksikön tilatiedon välittömästi. Yksikön tilatietoja tulisi voida tarkastella myös tehtävänäyttöikkunan kautta.

#### **4.1.2 Virheiden ja erikoistapausten sieto**

Sovelluksen pitää sietää häiriöitä ja virhetilanteita. Virhetilanteita voivat aiheuttaa puuttuvat PEKE-viestit tai korruptoitunut tieto. Toisinaan TETRA-verkko kadottaa viestejä kokonaan, jolloin tehtävästä saatava informaatio jää vajaaksi. Sovellus täytyy kehittää siten, että jo ensimmäiseen viestiin reagoidaan ja sen sisältämä tieto näytetään välittömästi ja myöhemmin saapuvat viestit täydentävät näitä tietoja. Toisinaan tehtäväviestit eivät sisällä kaikkea niiden mahdollistamaa tietoa. Tuolloin on huomioitava, että viestikentät saattavat olla tyhjiä tai puutteellisia.

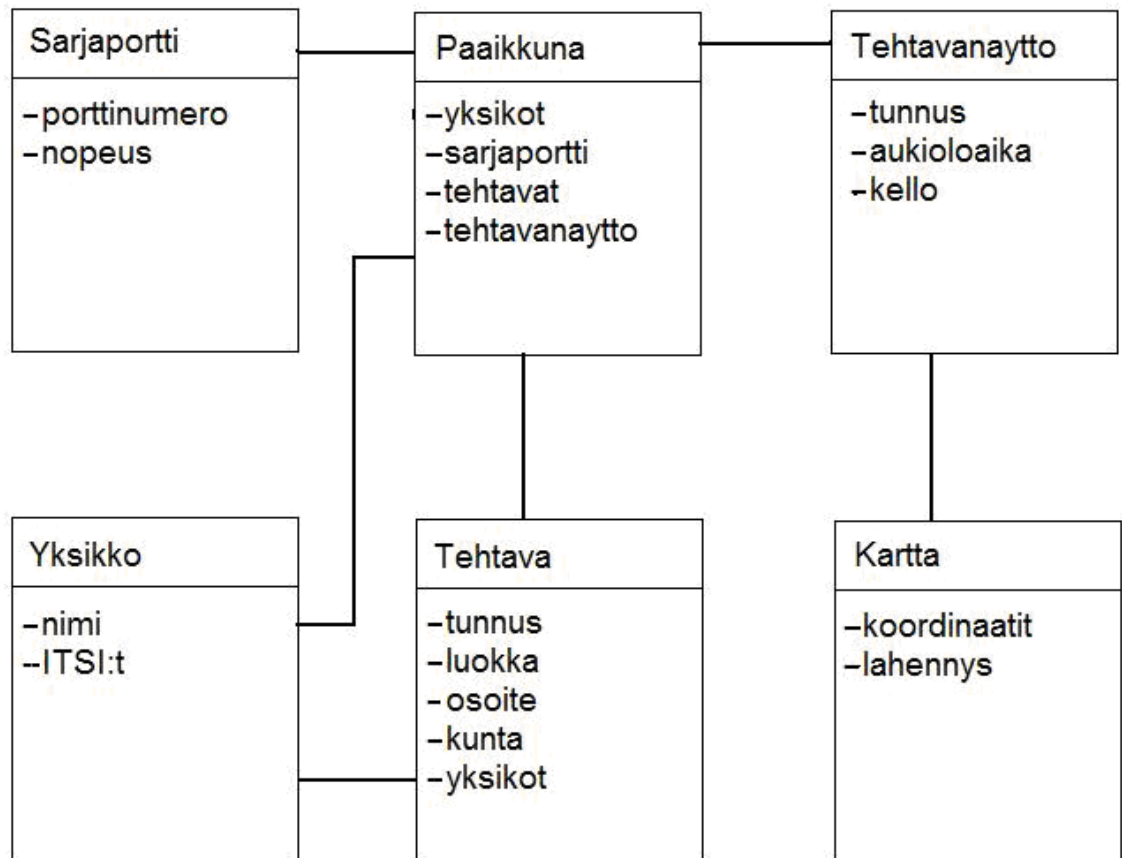
On mahdollista, että tehtäviä tulee lyhyen ajan sisään useita, mikä voi aiheuttaa ongelmia viestien vastaanotossa ja esittämisessä. Sovelluksessa asia ratkaistaan mahdollistamalla tehtävien selaaminen edestakaisin aikajärjestyksessä. Näin ollen lyhyen ajan sisällä tulleet tehtävät eivät aukaise ikkunoita päällekkäin, vaan uusimman tehtävän tiedot näkyvät päällimmäisenä ja vanhempia tehtäviä voi selata nuolinappien avulla.

Sovellus on aina päällä, joten muistin käyttö ja palautus on tärkeää. Sovelluksen pitää olla käyttövarma ja kaatumisia aiheuttavat ohjelmointivirheet täytyy eliminoida pois. Erikoistilanteet, kuten kaikkien tehtävien tai yksiköiden poistaminen, eivät saa aiheuttaa virhetilanteita listauksissa. Jos tehtävään on kiinnitetty tietokantaan kirjaamaton yksikkö, on se sinne lisättävä automaattisesti, jotta vältetään virhetilanteilta. Näin myös yksiköiden listaus pysyy automaattisesti ajan tasalla.

## 4.2 Luokkakaavio

Sovellus tehdään olio-ohjelmointia hyödyntäen, eli se koostuu luokista ja olioista. Pääikkuna-luokka sisältää muutamia luokkia, kuten tehtavanaytto-, sarjaportti-, tehtava- ja yksikko-luokkia. Luokkakaavio on alustava suunnitteluvaiheen luokkakaavio, joten se tulee muuttumaan ja tarkentumaan myöhemmin toteutusvaiheessa. Esimerkiksi muuttujien tietotyyppejä tai nimiä ei ole vielä lyöty lukkoon. Samasta syystä luokkien funktioita ei ole tarkemmin määriteltä.

Tehtavanaytto-luokka edustaa tehtävänäyttöikkunaa, joten se sisältää myös käyttöliittymäkomponentteja. Se sisältää myös kartta-luokan, joka tulee hoitamaan karttaan liittyviä toimintoja. Yksiköt-luokka esittää yksiköitä, ja se sisältää nimen ja ITSI:t. Tehtava-luokka kuvaa tehtäviä. Ne sisältävät perustietoja tehtävään liittyen ja lisäksi tehtävään kiinnitetyt yksiköt. Tästä johtuu kuviossa 5 näkyvä assosiaatio luokkien välillä. Sarjaportti-luokka hoitaa sarjaporttiliikenteeseen kuuluvat tehtävät.



KUVIO 5. Suunnitteluvaiheen luokkakaavio

### 4.3 Tietokanta

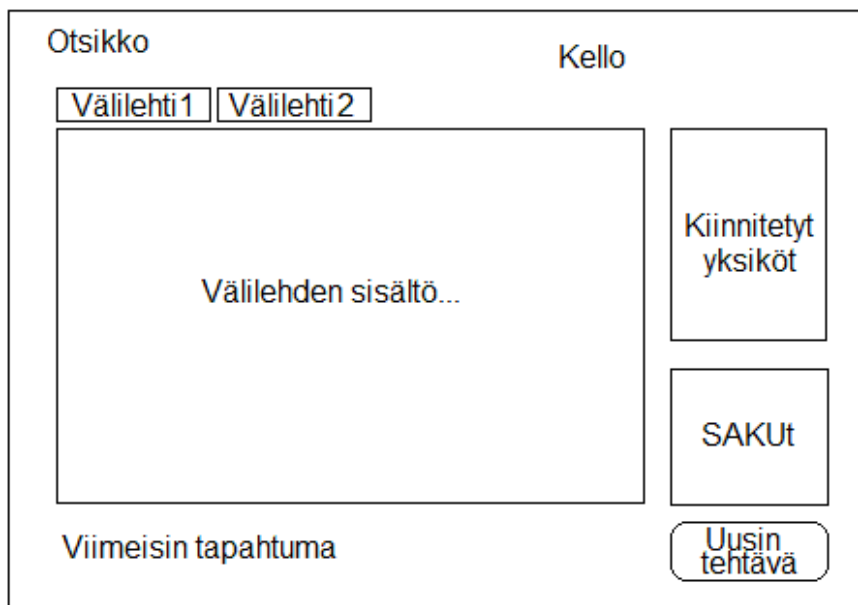
Ohjelman tietovarastona käytetään MySql-tietokantaa, jonne varastoidaan tarvittavat tiedot. Jos tietokanta sijoitetaan erilliselle palvelimelle, joka on yhteydessä internetiin, voi tehtäviä ja yksiköiden tiloja tarkastella myös ilman PEI-rajapintaa ja TETRA-päätelaitetta.

Tietokantaa ei ole tarkoitettu arkistoimaan tietoja tehtävistä tai yksiköistä. Tietoturvaisyistä tietokannassa säilytetään tiedot vain muutamia viikkoja asetusten mukaan. Tietokantaa käytetään siis vain väliaikaiseen tallentamiseen ja tietojen palauttamiseen esimerkiksi sähkökatkon tai sovelluksen uudelleen käynnistämisen jälkeen. Liitteessä 2 on esitetty tietokannan ja sen taulujen rakenne kaaviona ja selitetty rakennetta tarkemmin sanallisesti.

#### 4.4 Käyttöliittymä

Asiakkaan toivomuksesta päädyttiin käyttöliittymään, jossa on kaksi ikkunaa: pääikkuna ja tehtäväikkuna. Pääikkunaan ryhmitellään ohjelman toiminnot välilehtiin. Tehtäväikkunaan sijoitetaan yksittäiseen tehtävään liittyvät tiedot. Pääikkuna tulee sisältämään kaikki ohjelman taustatoiminnot ja asetukset. Kuten on aiemmin mainittu, tehtävänäyttöikkuna ilmestyy automaattisesti päällimmäiseksi tietokoneen näytölle tehtäväviestien saapuessa.

Pääikkuna jaetaan välilehtiin, joihin on ryhmitelty ohjelman toiminnot. Esimerkiksi asetukset ja viestien lähetys ovat eri välilehdissä. Osa tiedoista näkyy pääikkunassa kuitenkin koko ajan, riippumatta valitusta välilehdestä. Ylimmäisenä on otsikko, joka kertoo lyhyesti ohjelmasta. Se voisi olla esimerkiksi: "Kokkolan palolaitos PATI-tehtävänäyttö". Otsikon viereen sijoitetaan kello, joka kertoo ajan ja päivämäärän. Kellon alapuolelle sovitetaan laatikko, johon kiinnitetyt yksiköt ilmaantuvat. Tämän alle tulee toinen laatikko, jossa ovat kiinnitetyt sairaankuljetusyksiköt. Pääikkunassa alimmaiseksi laitetaan tilarivi, jossa kerrotaan viimeisin tapahtuma. Tapahtuma voi olla esimerkiksi yksikön tilan muuttuminen tai tehtäväviestin saapuminen. Tilarivistä vasemmalle, nurkkaan sijoitetaan painike, josta saa näkyviin viimeisimmän tehtävän tiedot. Kuviossa 6 on havainnollistettu pääikkunan rakenne ja tärkeimmät komponentit.



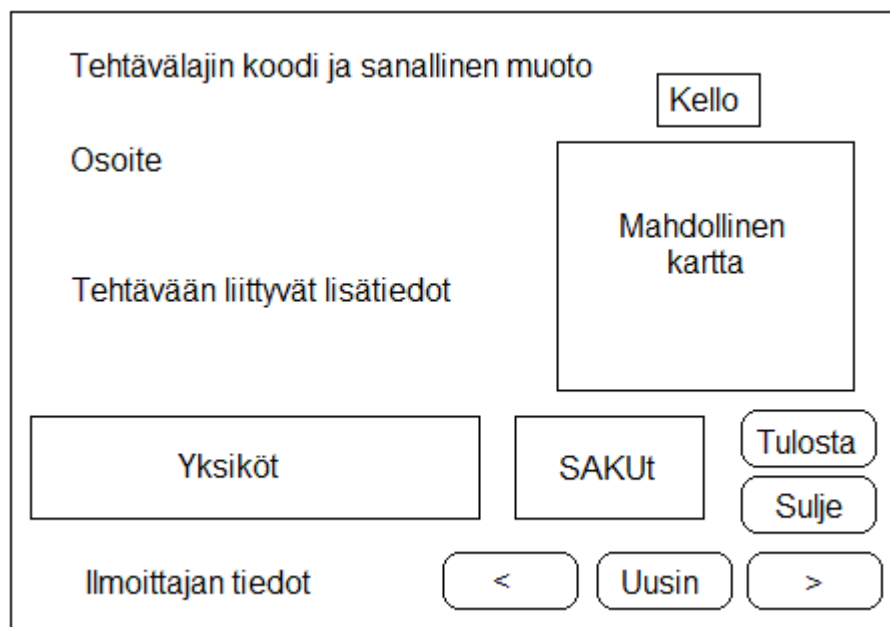
KUVIO 6. Pääikkunan käyttöliittymä

Tehtävänäyttöikkunan käyttöliittymä koostuu kartasta, painikkeista ja tietokentistä. Tiedot tulevat tärkeysjärjestykseen, ja tärkeyttä korostetaan tekstin koolla. Ensimmäinen tietokenttä kertoo tehtävälajin koodina ja sanallisessa muodossa. Tehtäväkoodin jälkeen voi olla kiireellisyystieto, ilmaistuna kirjaimilla A–D. Näiden alle laitetaan osoitetiedot. Osoitetiedot koostuvat yhdestä tai kahdesta kadun nimestä, talonumerosta ja kunnasta. Osoitetietojen jälkeen tulevat tehtävään liittyvät lisätiedot. Hätäkeskuksen päivystäjältä riippuen tämä tietokenttä sisältää tehtävään liittyviä tietoja tai ilmoittajan antamia tietoja.

Lisätietojen alle tulee kaksi laatikkoa, jotka sisältävät painikkeita. Ne kuvastavat yksiköitä. Vasemmanpuoleinen laatikko sisältää tehtävään kiinnitetyt yksiköt. Oikeanpuoleisessa laatikossa ovat kiinnitetyt sairaankuljetusyksiköt. Painikkeiden niminä ovat yksiköiden nimet ja taustaväri on yksikön sen hetkinen tilatieto. Tilatieto päivittyy painikkeiden taustaväreiksi automaattisesti ja reaaliaikaisesti. Kiinnitetyt sairaankuljetusyksiköt kertovat nopeasti ja helposti saatavilla olevat ambulanssit. Painikkeita painamalla aukeaa ponnahdusikkuna, jossa on kyseessä olevan yksikön tilatieto sanallisesti ja taustaväriä esitetty.

Yksiköiden alapuolella ovat ilmoittajan yhteystiedot ja muutamia painikkeita. Painikkeilla on mahdollista tulostaa paperille tehtävän tiedot, sulkea ikkunan tai selata tehtäviä aikajärjestyksessä edestakaisin. Tämä toiminto on tarpeellinen päällekkäisten tehtävien sattuessa. Uusin-painikkeella tehtävänäyttöikkunaan saa uusimman tehtävän tiedot. Kuviossa 7 on kuvattu tehtävänäyttöikkunan rakenne.

Kartta ja kello sijoitetaan oikealle. Kello ei ilmaise sen hetkistä aikaa, vaan se on sekuntikello, joka kertoo viimeisimmän tehtävän antohetkestä kuluneen ajan. Kartta kohdistetaan viesteissä annettujen koordinaattien perusteella ja siihen merkitään kohde nastalla.



KUVIO 7. Tehtävänäyttöikkunan rakenne ja komponentit



## 5 TOTEUTUS

Kuten aikaisemmin on mainittu, tehtävänäyttösovellus ohjelmoitiin C++:lla käyttäen hyväksi Qt:n tarjoamia kirjastoja. Tietokantana käytettiin MySQL:ää. Tietoliikenteessä käytettiin TETRA-verkkoa ja -päätelaitetta. Ohjelmaa ajetaan Windows-käyttöjärjestelmällä ja ohjelmaa käytetään kosketusnäytön avulla.

### 5.1 Käytetyt Qt-luokat

Ohjelmiston kehityksessä käytetyistä Qt:n tarjoamista luokista tärkeimpinä voidaan mainita käyttöliittymään liittyvät luokat ja QWebKit-luokka. Käyttöliittymän toteuttamiseen käytettiin seuraavia luokkia: QPushButton-, QLabel-, QImage-, QGroupBox-, QTabWidget-, QTableWidget-, QRadioButton-, QCheckBox-, QLineEdit-, QTimeEdit-, QTextEdit- ja QSpinBox-luokka. Qt tarjoaakin lähes jokaiseen tarpeeseen valmiin käyttöliittymäkomponentin, joten käyttöliittymän toteutus oli nopeaa ja helppoa.

QWebKit tuo Qt:lle todella paljon lisäarvoa. Se mahdollistaa www-sivujen esittämisen ohjelman sisällä. Tämä lisää ohjelmien monimuotoisuutta ja helpottaa ohjelmistokehittäjien työtä. QWebKit toimii aivan kuin internetselaimena ohjelman sisällä. WebKitiä käyttämällä kartan upottaminen ohjemaan oli yksinkertaista.

Myös ohjelman toimintojen toteutukseen käytettiin laajasti hyväksi Qt-luokkia. Tärkeimpänä käytössä oli QString-luokka, jonka avulla tekstimuotoista dataa voi muokata ja käyttää. Myös ajastinta eli QTimer-luokkaa hyödynnettiin ja muita aikaan liittyviä, kuten esimerkiksi QDate tai QTime.

## 5.2 MySql-tietokanta

Qt ei tue MySql:ää suoraan, vaan se on käännettävä suoraan lähdekoodista lisäosaksi Qt:hen. Ensiksi on ladattava tarvittavat lähdekoodit. Koodit ovat saatavilla ilmaiseksi ja luvallisesti, koska MySql on niin sanotusti vapaalla lähdekoodilla toteutettu.

Seuraavassa vaiheessa on asetettava configure-ohjemalla Sql-lisäosaksi MySql. Tämä tapahtuu käytännössä Qt:n omalla komentorivillä komennolla: *configure --plugin-sql-mysql*. Kolmannessa vaiheessa käännetään MySql:n lähdekoodi Qt:lle, jolloin saadaan Qt:lle uusimmat ajurit tietokanta komentoja varten. Kääntäminen tapahtuu komentoriviltä komennolla: *qmake -o Makefile "INCLUDEPATH+=C:\MySQL\include" "LIBS+=C:\MySQL\lib\opt\libmysql.lib" mysql.pro* ja lopuksi ajetaan kääntäjä: *nmake*.

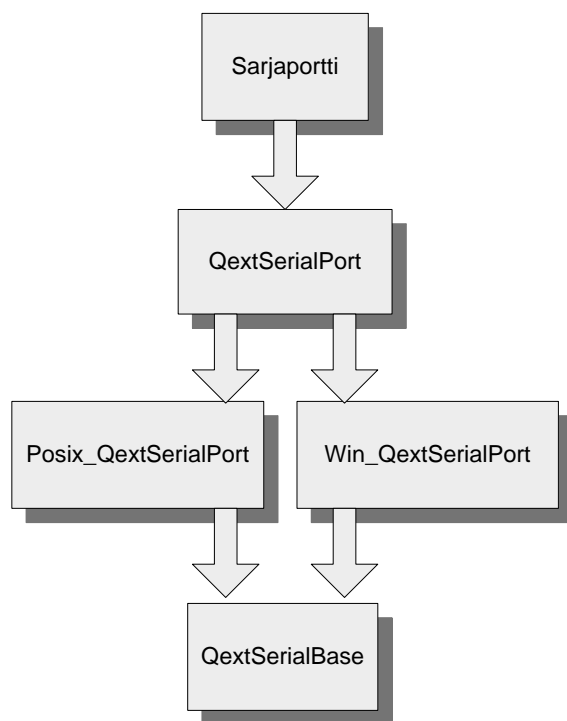
Toimenpiteet vaativat aikaa runsaasti, mutta se on sen arvoista. Qt:n mukana tulee tuki SQLite:lle, mutta se ei ole käytettävyydeltään tai tietoturvaltaan verrattavissa MySql:ään. MySql-tietokanta voidaan myös QqLite:stä poiketen sijoittaa turvallisesti myös erilliselle internetpalvelimelle.

## 5.3 Sarjaporttiyhteys

Sarjaporttiyhteyden luominen TETRA-puhelimen ja ohjelman välille oli tärkein osa ohjelman kehitystä. Tehtävänäyttösovelluksessa hyödynnettiin GPL-lisenssin alaista lähdekoodia, joten sarjaportin ohjausta ei ryhdytty itse ohjelmoimaan alusta saakka. Sarjaporttiyhteys on toteutettu muutamalla luokalla. Windows- ja Linux-käyttöjärjestelmiä varten on kehitetty erilliset luokat ja rajapinnat, mutta ylemmät luokat ovat samoja molemmissa käyttöjärjestelmissä. Ohjelmointi on tällöin yksinkertaista. Ohjelma on vain käännettävä erikseen eri alustoille.

Kuten kuviossa 8 näkyy, ManageSerialPort on korkein luokka. Se tarjoaa käyttökelpoisen rajapinnan sarjaportin ja ohjelman välille. Esimerkiksi ohjelmoijalla

on käytössä signaali, joka emittoituu, kun sarjaportissa on tulossa dataa. Seuraavat kaksi tasoa hoitavat kommunikoinnin laitteiston kanssa, eli ne ovat yhteydessä fyysiseen sarjaporttiin.



KUVIO 8. Sarjaporttirajapinnan luokat

TETRA-puhelimien asetuksia on mahdollista muuttaa AT-komentojen avulla. Tehdasasetuksissa on sarjaporttiliikennettä varten asetettu oletusasetukset, joita ei lähde muokkaamaan, vaan ne on otettu huomioon ohjelmoitaessa sarjaporttirajapintaa. Tärkeimmät asetukset tehtävänäytön toimivuuden kannalta ovat käytetty merkitö, PEI-liittymän asetukset ja vuon ohjaus. Merkitöksi valittiin Latin1-merkitö, jota tukevat sekä Qt että käytetty päätelaite.

Käynnistuksen yhteydessä sovellus asettaa päätelaitteeseen tarvittavat asetukset automaattisesti ja tarkastaa yhteyden toimivuuden. Yhteyden toimivuus todetaan lukemalla sarjaporttiliikennettä kymmenen sekuntia, jonka aikana liikennettä on esiinnyttävä, jotta voidaan todeta yhteyden toimivan. Voidaan olettaa, että paikannuksesta johtuvaa tietoliikennettä esiintyy kymmenen sekunnin aikana useita kertoja.

## 5.4 Saapuvan datan käsittely

Sarjaportti-rajapinta tarjoaa signaalin, joka emittoituu, kun sarjaporttiin tulee ulkoapäin tietoa. Saapuva signaali yhdistetään metodiin, joka ajetaan. Metodi suorittaa datalle tarvittavat toimenpiteet. Sarjaporttiin saapuva data koostuu hätäkeskuksen lähettämistä tehtäväviesteistä, paikkatiedoista ja tilatiedoista. Kaikki sarjaportista tuleva tieto prosessoidaan ja otetaan talteen. Prosessoidusta tiedosta poimitaan asianmukaiset tiedot jatkokäsittelyyn. Toisin sanoen sarjaporttiin saapuvasta tiedosta poimitaan tehtäväviestit ja tilatiedot erilleen muusta datasta. Viestit yhdistellään tehtävänumeron perusteella kokonaisuudeksi, joka esitetään tehtävänäyttöikkunassa.

Saapuneista tilatiedoista poimitaan lähettäjän ITSI ja annettu tilatieto. Näiden avulla kyseiseen ITSliin kiinnitetyn yksikön tila muutetaan tilatietoa vastaavaksi. Esimerkiksi, jos kyseinen yksikkö on nähtävillä kiinnitettynä tehtävään tai sairaankuljetusajoneuvoksi, sen väri muuttuu tilaa vastaavaksi. Jos kiinnitettyä yksikköä ei ole määritetty tietokannassa, ohjelma lisää sen sinne.

## 5.5 Kartta

Kartan yhdistäminen tehtävänäyttöikkunaan toteutettiin helpointa mahdollista ratkaisua käyttäen. Tehtävänäyttöikkunaan liitettiin QWebKit-elementti, joka mahdollistaa internetsivujen esittämisen upotettuna ohjelmaan. QWebKitissä ajettiin JavaScriptiä, jolla karttapohjat toteutettiin hyödyntämällä paikannus- ja karttapalveluita tarjoavia rajapintoja, kuten Google Mapsia tai OpenStreetMapia.

Kartalla pyrittiin helpottamaan ja nopeuttamaan kohteen osoitteen hahmottamista. Kartta kohdistuu annettujen koordinaattien perusteella. tehtäväviesteissä koordinaatit on annettu karttakoordinaatistojärjestelmän (KKJ) mukaisina koordinaatteina. Kohde on merkitty kartalle rastilla.

### 5.5.1 Google Maps ja OpenStreetMap

Googlen tarjoama rajapinta mahdollistaa Googlen tarjoamien karttojen ja ilmakuvien tehokkaan ja helpon hyödyntämisen. Karttoja voidaan kohdistaa koordinaattien avulla, ja niille voi myös lisätä nuppineuloja ja tietokuplia. Googlen tarjoaman rajapinnan käyttöön tarvittava JavaScript-koodi on lyhyt ja yksinkertainen:

```
var map;  
var marker = new GMarker(new GLatLng(0,0));  
function initialize()  
{  
    if (GBrowserIsCompatible())  
    {  
        map = new GMap2(document.getElementById("map"));  
        map.setCenter( new GLatLng(0, 0), 1);  
    }  
}  
function Open(x,y,z)  
{  
    map.setCenter( new GLatLng(x, y), z);  
    marker.setLatLng(new GLatLng(x, y));  
    map.addOverlay(marker);  
}
```

Aluksi alustetaan tarvittavat muuttujat. Tässä tapauksessa muuttujina ovat kartta (map) ja nuppineula (marker). Kun JavaScript ajetaan selaimessa, ensimmäiseksi suoritetaan initialize-funktio. Funktiossa alustetaan kartta ja asetetaan sijainti ja muut mahdolliset määrittelyt.

Funktiota Open käytetään, kun halutaan kohdistaa kartta tiettyyn pisteeseen. Samalla nuppineulalle annetaan uudet koordinaatit, jolloin se sijoittuu kartan keskelle. Funktiolle annettavat muuttujat x, y sisältävät WGS84-koordinaatiston mukaiset koordinaatit tunteina ilmaistuna. Muuttuja z ilmoittaa halutun lähen-nyksen.

PEKE-tehtäväviestien KKJ-koordinaattien eroaminen kansainvälisesti käytetystä WGS84-koordinaateista aiheutti ongelmia. Koordinaattien muuntaminen järjestelmästä toiseen on varsin monimutkainen ja aikaa vievä prosessi. Toimivan muunnosfunktion toteuttaminen itse ei ollut prototyyppivaiheessa järkevää, joten ohjelmassa hyödynnettiin jo olemassa olevia palveluita. Esimerkiksi Maanmittauslaitoksen Kansalaisen karttapaikka -internetsivustossa voi suorittaa koordinaattimuutoksia.

Vaihtoehtona Googlen tuotteelle on täysin ilmainen OpenStreetMap-rajapinta, joka tarjoaa lähes samat palvelut. OpenStreetMap perustuu käyttäjien itse tekemiin karttoihin. Ongelmaksi muodostuu tällöin karttojen luotettavuus ja kattavuus. Edellä mainittujen riskitekijöiden vuoksi OpenStreetMap ei ollut varten-otettava vaihtoehto karttapohjaksi.

### 5.5.2 Ongelmat

QWebKitin, JavaScriptin ja Google Mapsin käyttö kartan toteuttamiseen osoit-tautui vasta testausvaiheessa ongelmalliseksi. Ensinnäkin Google Mapsin käyt-tö ilman erillistä korvausta suljetuissa järjestelmissä on kielletty käyttöehdoissa. Toiseksi pitempiaikaisessa käytössä ilmeni muistivuoto-ongelmia. Muistivuoto aiheutti ohjelman hidastumista etenkin karttaa liikutettaessa ja tarkennettaessa.

Muistivuodon tarkkaa aiheuttajaa tai syytä ei pystytty testattaessa kohdenta-maan tarkasti. Kartan ja QWebKitin poistaminen sovelluksesta eliminoi ongel-man täysin. JavaScriptin ja Google Mapsin käyttö on aiheuttanut ongelmia ai-kaisemminkin, ja muistivuotoja on korjattu eri selaimille useaan kertaan. QWebKit on suhteellisen uusi tuote, joten kehitettävää riittää.

Kartan toteuttaminen muillakin keinoin on tietenkin mahdollista, mutta aikaa vievää ja monimutkaista. Karttapohjat ovat poikkeuksetta maksullisia ja toimivan karttamoottorin toteuttaminen prototyyppiin ei ole järkevää. Edellä mainituista syistä kartta otettiin kokonaan pois käyttöliittymästä.

## 5.6 Testaus

Testausta toteutettiin toistuvasti ohjelmoinnin edetessä. Ohjelman osat ohjelmoitiin tärkeysjärjestyksessä. Aluksi testattiin sarjaporttiliikenteen vaatimat luokat, seuraavaksi viestien käsittely- ja yhdistämisrutiinit ja lopuksi käyttöliittymään liittyvät hienosäädöt.

Testauksen alkuvaiheessa tärkeäksi osaksi muodostui sarjaportissa tapahtuvan liikenteen tarkastelu. Saatavilla ei ollut tarkkoja tietoja tai määritelmiä siitä, missä muodossa tai miten tieto välitetään. Esimerkiksi käytetyn päätelaitteen tapa siirtää vastaanotetut viestit sarjaporttiin tuotti ongelmia. Viestien saapuminen sarjaporttiin alkaa merkkijonolla ”+CTSDSR”, joka ilmoittaa, että viestin lähetys sarjaporttiin alkaa. Seuraavaksi sarjaporttiin saapuvat otsikkotiedot. Lopuksi ei tule mitään merkkiä tai merkkijonoa, joka ilmoittaisi viestin loppuneen. Ongelmaksi muodostui se, ettei voida tietää, milloin viesti loppuu. Tämä ongelma voitiin sivuuttaa, koska sarjaportissa viestejä saapuu yleensä parikymmentä minuutissa, jolloin viestit voidaan erotella helposti otsikkotietojen avulla.

Testauksen aikana todettiin sarjaporttiliittymän kuluttavan runsaasti prosessori-aikaa ja -tehoa. Ongelma johtuu sarjaporttiluokan heikosta toteutuksesta. Saapuvan datan toteamiseen ei käytetä keskeytyksiä, vaan porttia luetaan jatkuvasti päättymättömässä silmukassa. Tämä aiheutti käytössä olleessa laitteistossa merkittävän korkeaa prosessorin kuormitusastetta.

Sarjaporttiliikenteen toimivuuden toteamisen jälkeen testaukseen tuli saapuvan datan seulominen ja yhdisteleminen. Saapuvat tehtäväviestit tunnistetaan vies-

teissä aina toistuvien yksityiskohtien perusteella, joita ovat tietokenttien erotinmerkki ja lukumäärä tai kentän tietotyyppi. Toisinaan sarjaporttiin tulleet paikannustiedot menivät seulasta läpi, mutta ne eivät kuitenkaan voi sekoittua oikeisiin tehtävätietoihin tai -viesteihin. Näin ne eivät haitanneet merkittävästi ohjelman toimintaa. Tämä ominaisuus antaa pienen mahdollisuuden häiritä ohjelman toimintaa lähettämällä tehtäväviestien määritelmiä noudattavia viestejä päälaitteeseen. Tämä tosin vaatii toimivan tetrapäätelaitteen, jonka ITSI tallentuisi kuitenkin ohjelman lokiin.

Käyttöliittymän testaamiseen käytettiin pelastuslaitoksen omaa henkilöstöä. Testauksen yhteydessä tuli ohjelman käytettävyydestä tärkeää tietoa, ja palautteen perusteella toteutettiin muutamia uusia ominaisuuksia. Käytettävyyttä lisättiin painikkeiden uusilla nimillä ja sijoittelulla. Uutena ominaisuutena tuotettiin sairaankuljetusyksiköiden erottaminen tavallisista yksiköistä ja yksiköiden sijoittelu asemien ja paikkakuntien mukaan.

Testauksen ja ohjelman edetessä testattiin myös ohjelman luotettavuutta ja toimivuutta. Kartan aiheuttamat kaatumiset todettiin jo varhaisessa vaiheessa. Käytetty päätelaite aiheutti myös ongelmia. Muutamia kertoja puhelimen asetukset nolloutuivat tai puhelin uudelleenkäynnistyi itsekseen. Päätelaitteissa esiintyviä ongelmia ei ryhdytty tutkimaan opinnäytetyön puitteissa. Kun ongelma oli todettu, laite vaihdettiin.

Ohjelman käyttötuntien kasvaessa huomioitiin tarpeelliseksi muutama lisätoiminto. Tietoturvasyistä vanhat tehtävät täytyy poistaa tietokannasta ja ohjelman muistista automaattisesti, ilman käyttäjän toimenpiteitä. Vaikeiden valaistusolosuhteiden vuoksi sovelluksen väritystä muutettiin kontrastin lisäämiseksi taustan ja tekstin välille. Ikkunan oletustaustaväri harmaa muutettiin valkoiseksi.

## **5.7 Kuvia valmiista tehtävänäyttösovelluksesta**

Kuviossa 9 näkyy tehtävänäyttösovelluksen pääikkuna. Välilehdistä on aktiivisena palolaitokset-välilehti, jossa on valitut yksiköt lajiteltu paloasemien mu-





kaan. Kuvassa näkyvät värilliset laatikot kuvaavat yksiköitä ja niiden tilatietoja. Oikeassa ylänurkassa näkyvä hiirikuvake ei ole ohjelman osa vaan se liittyy kosketusnäytön ohjaamiseen.



Kuvio 9. Palolaitokset-välilehti

Kuviossa 10 on valittuna yksiköt-välilehti. Välilehdessä näkyvässä taulukossa on yksiköt ja niiden tärkeimmät tiedot. Tilatiedot ovat sanallisesti esitettyinä, ja myös niiden tapahtuma-aika on näkyvillä. Taulukon otsikoita painelemalla ne järjestyvät kyseessä olevan sarakkeen mukaisesti joko ajalliseen tai aakkoselliseen järjestykseen. Painikkeiden avulla yksiköitä ja ITSEjä voi poistaa ja lisätä. Välilehden vasemman puoleisimmilla napeilla voidaan lisätä ja poistaa kiinnitettyjä yksiköitä. Kuviossa 11 on valittuna tehtävät-välilehti. Tehtävät on taulukoitu ja vain tärkeimmät tiedot ovat näkyvissä. Tietoturvasyistä tehtävien tiedot on pikselöity.

 **PATI-tehtävänäyttö: Kokkolan palolaitos** 25.11.2010 14:58 

Asetukset / Lohi / Lähetä viestejä / Testaus / Yksiköt / Tehtävät / Ohje / Palolaitokset

Avaa

Lisää ITSI

Poista ITSI:t

Lisää yksikkö

Poista yksikkö

	Nimi	ITSI	Pvm	Tila	Kommentti
40	K182	2272682		EI TILATIE TOA	
41	K191	5272911, 527...	2010.11.24 1...	Asemalla	
42	K192	5272921, 527...	2010.11.23 0...	Asemalla	
43	K193	8, 5272930	2010.09.09 1...	Ei Hälyt	
44	K194	9, 5272940	2010.07.30 1...	Ei Hälyt	
45	K198	1, 5236981	2010.11.24 2...	Asemalla	
46	K21	2272221	2010.11.25 1...	ASEMAPAIK...	
47	K27	2272227		EI TILATIE TOA	
48	K3	2272103	2010.11.25 1...	ASEMAPAIK...	
49	K32	2272340		EI TILATIE TOA	
50	K35	2272350		EI TILATIE TOA	
51	K41	2272411	2010.05.19 1...	ASEMAPAIK...	
52	K51	2315111	2010.11.16 1...	ASEMAPAIK...	
53	K53	2315113	2010.11.09 1...	VAPAA	
54	K57	2315117	2010.11.18 1...	ASEMAPAIK...	
55	K591	5315191	2010.11.25 1...	KOhteessa	
56	K61	2428111	2010.11.19 0...	ASEMAPAIK...	



25.11. 14:57:28  
Tilatiesto saapui: 5429911

**Uusin tehtävä**

**Sairaankuljetus**

K191 K192 K193  
K194 K198

KUVIO 10. Yksiköt-välilehti

 **PATI-tehtävänäyttö: Kokkolan palolaitos** 25.11.2010 15:07 

Asetukset / Lohi / Lähetä viestejä / Testaus / Yksiköt / Tehtävät / Ohje / Palolaitokset

Avaa

Poista

	ID	Laji	Pvm	Osoite	Kommentti
3	K182	K182	2010.11.24 1...	2010.11.24 1...	
4	K191	K191	2010.11.24 1...	2010.11.24 1...	
5	K192	K192	2010.11.23 0...	2010.11.23 0...	
6	K193	K193	2010.09.09 1...	2010.09.09 1...	
7	K194	K194	2010.07.30 1...	2010.07.30 1...	
8	K198	K198	2010.11.24 2...	2010.11.24 2...	
9	K21	K21	2010.11.25 1...	2010.11.25 1...	
10	K27	K27			
11	K3	K3	2010.11.25 1...	2010.11.25 1...	
12	K32	K32			
13	K35	K35			
14	K41	K41	2010.05.19 1...	2010.05.19 1...	
15	K51	K51	2010.11.16 1...	2010.11.16 1...	
16	K53	K53	2010.11.09 1...	2010.11.09 1...	
17	K57	K57	2010.11.18 1...	2010.11.18 1...	
18	K591	K591	2010.11.25 1...	2010.11.25 1...	
19	K61	K61	2010.11.19 0...	2010.11.19 0...	

25.11. 15:07:03  
Tilatiesto saapui: 5233291

**Uusin tehtävä**

**Sairaankuljetus**

K191 K192 K193  
K194 K198

KUVIO 11. Tehtävät-välilehti

Kuviossa 12 on kuvaruutukaappaus tehtävänäyttöikkunasta. Kuvasta näkyy tietojen ja painikkeiden sijoittelu. Osa tiedoista on sumennettu tietoturvasyistä. Kartalle varattuun paikkaan on laitettu valmiissa versiossa päivämäärä ja aika.

202C

tieliik.onnett.: pieni

KATU ,

KATU, Kokkola

2010-11-25

2 h-auto, kylkikolari, 1 autolija *kipua*, 11 autoilija *kipu*, *haittaa liikennettä*

Yksiköt

K3

K11

K131

K132

K15

K191

Sairaankuljetus

K191

K192

K193

K194

K198

9

◀

Uusin

▶

Tulosta

Sulje

KUVIO 12. Tehtävänäyttöikkuna

## 6 POHDINTA

Opinnäytetyön tuotoksena syntynyt sovellus on prototyypiluonteestaan huolimatta jatkuvassa käytössä. Ulkonäöltään ja toiminnallisuuksiltaan se toimisi myös tuotteistettuna versiona. Ohjelma on yksinkertaisuutensa vuoksi luotettava ja toimiva kokonaisuus, vaikka kehitettävää varmasti löytyy.

Tehtävänäyttösovellus osoittautui varsin toimivaksi ratkaisuksi tärkeimmällä toiminta-alueellaan. Tehtävien vastaanotto ja näyttö sujuu kitkattomasti ja nopeasti. Viive viestien saapumisen ja näytölle ilmestymisen välillä on olematon.

Tehtävänäyttösovellusta on mahdollista käyttää myös muissa Suomen pelastuslaitoksissa. Ohjelman tarjoama yhteys TETRA-puhelimen ja tietokoneen välillä olisi myös hyödynnettävissä yritysmaailmassa. Esimerkiksi suurissa tuotanto- tai voimalaitoksissa on myös käytössä TETRA-verkkoja.

Ohjelman toteutus käyttäen Qt-luokkia oli perusteltua ohjelman siirrettävyyden kannalta. Lähdekoodista Linux-palvelimelle tai mobiilialustoille kääntäminen on pienin muutoksin mahdollista ja helppoa. Mahdolliset mobiilisovellukset on helppo toteuttaa käyttäen erillistä internetpalvelinta. Käyttö rajoittuu kuitenkin Symbian 60 sarja-, Maemo- tai Meego-alustaiseen laitteeseen. Palvelimella olevaa tietokantaa olisi kuitenkin päivitettävä tietokoneella olevalla tehtävänäyttösovelluksella. Mobiilisovellus kuitenkin vapauttaisi käyttäjän PEI-liittymästä ja TETRA-päätelaitteesta.

Ohjelmiston kehittäminen oli oppimisprosessi. Prosessin edetessä Qt-ohjelmointitaidot karttuivat ja sekä lähdekoodin luettavuus että tehokkuus kasvoivat. Sovelluksen ohjelmointi vaati jatkuvaa opiskelua ja tutkimusta. Opinnäytetyön tekeminen tuki merkittävästi opiskelua ja mahdollisti Qt:n syvällisemmän tutkimisen. Koko ohjelmointiprosessin läpikäyminen yksin, suunnittelusta toteutukseen ja testaukseen, oli mielenkiintoista ja kannustavaa. Tämä mahdollisti myös omaperäiset ratkaisut ja ulkoasun ohjelmaan.

Sarjaporttiliikenne ja standardit hahmottuivat aikaisempaa syvällisemmin, ja niihin sai perehtyä perusteellisesti. Pelastustoimi oli alana mielenkiintoinen ja sen käyttämät TETRA-laitteet mielenkiintoisia ja ainutlaatuisia. Työ mahdollisti myös tutustumisen yleisesti pelastustoimeen ja hälytystehtävien sisältöihin.

Ohjelman tekeminen täysin itsenäisesti ja väljin ohjein oli mukavaa ja opettavaista. Suunnitteluvaiheen työläys kuitenkin yllätti ja toteutusvaiheen ongelmat olivat monimutkaisia ja aikaa vieviä. Ohjelman kehitys ei edennyt täysin aikataulun mukaan, mikä johtui osin toteutusvaiheen ongelmista. Ongelmat toivat kuitenkin haastetta, mutta se oli tarpeellista mielenkiinnon säilymiseksi.

Pelastustoimi on yhteiskunnallisesti tärkeä, ja sitä on pidetty varmana ja luotettavana. On tärkeää, että sitä kehitetään jatkuvasti ja uusinta teknologiaa hyödynnetään ennakkoluulottomasti edistämään yleistä turvallisuutta. TETRA-verkko tuo paljon mahdollisuuksia soveltaa tietotekniikkaa kehittävällä ja uudella tavalla myös pelastustoimessa. Ohjelmaa oli mukava tehdä, koska se on varmasti hyödyllinen ja tarpeellinen. Se myös auttaa tietoteknillistämään pelastustoimen toimintaa. Toivottavasti se myös tuo uusia ideoita ja tietoteknisiä sovelluksia, jotka auttaisivat pelastustoimea kehittymään entistä tehokkaammaksi ja luotettavammaksi.

## LÄHTEET

Blanchette, J. & Summerfield, M. 2009. C++ GUI Programming with Qt 4. Second Edition. 4. painos. Massachusetts: PRENTICE HALL.

Heikkonen, K., Pesonen, T. & Saaristo, T. 2005. VIRVE-radio. TETRA viranomaiskäytössä. Helsinki: Edita.

Nokia 2009. Qt Creator Whitepaper. Saatavissa: <http://qt.nokia.com/files/pdf/qt-creator-1.3-whitepaper> . Luettu 7.4.2010.

Nokia 2010. Qt Home. Saatavissa: <http://qt.nokia.com/> . Luettu: 23.3.2010.

Penttinen, J. 2006. Tietoliikennetekniikka. 3 G ja erityisverkot. Helsinki: WSOY.